Journal of Computers Vol. 29 No. 6, 2018, pp. 90-95
doi:10.3966/199115992018122906008

A Short Text Clustering Method Based on Deep Neural Network Model

Yan-li Liu, Xiao-jun Wen*

School of Computer Engineering, Shenzhen Polytechnic, Shenzhen, Guangdong 518055, China
wxjun@szpt.edu.cn

Received 1 June 2018; Revised 1 July 2018; Accepted 1 August 2018

Abstract. Aiming at the problem of short text clustering, this paper proposes a method to process text representations and clustering simultaneously. The method can be divided into two stages. In the first stage, the method uses the deep neural network to initialize parameters. In the second stage, the method also studies the text representation and clustering target through the deep neural network. The experimental results show that the proposed method achieves better results than the benchmark algorithm in a certain degree.

Keywords: clustering, deep neural network, short text

1 Introduction

Clustering, as an important data analysis and visualization tool, can be researched in the unsupervised learning domain from multiple perspectives. For instance, how to define a cluster center; how to choose the suitable distance measurement method; how to classify an instance into a class and so on. This paper focuses on clustering short text data. Short text is a common form of human communication, which is widely used in real-time chat systems, comments of social media networks and dialogue question-answering systems. The short text clustering method will help achieve the understanding of user-generated contents, and can be widely applied to the monitoring of public opinions and the construction of personalized emotional dialogue systems.

Different from ordinary text clustering, many words often appear only once in short texts, and the expression of words is relatively random. The sparseness and nonstandard feature of short texts make traditional text representations such as word frequency and term frequency–inverse document frequency method (TF-IDF) performs poorly in short text clustering. In order to solve these problems, some researchers focus on the study of using wikipedia [1] or ontology [2] to expand the context of short text data. However, these methods rely on an external knowledge base to achieve short text understanding and bring about two problems: First, limited by the size of the corpus of the knowledge base; Second, the use of external knowledge base to expand the short text will increase the memory consumption and the high-dimensional complexity of analysis processing. In order to avoid the problems, some scholars have tried to construct some complex models to achieve short text clustering. For example, Yin and Wang [3] proposed a dirichlet multinomial mixture model-based approach for short text. Cai et al. [4] proposed a Locality Preserving Indexing (LPI) algorithm for text clustering. However, how to design an effective model is an open-ended question. Most of the above methods use the Bag of Words model to achieve the understanding of short texts, and the ability to understand the semantics is relatively insufficient.

In recent years, with the rise of deep neural networks (DNN), many researchers have begun to use deep learning to learn features. For example, Hinton and Salakhutdinov [5] use a depth automatic encoder (DAE) to learn text representations. Moreover, with the development of word embedding [6], neural networks have shown their advantages in text representations, such as Recursive Neural Networks (RecNN) [7] and Recurrent Neural Networks (RNN) [8]. However, Recursive Neural Networks require high time complexity in constructing text trees, and Recurrent Neural Networks are a biased model. A

* Corresponding Author

convolutional neural network using convolution kernels to capture local features has achieved better results in many natural language processing tasks such as sentence modeling and relational classification. Therefore, this paper proposes an unsupervised text representation method using CNN.

After the text is expressed in a certain way, it needs to cluster the converted data. Traditional clustering methods, such as k-means, can use Euclidean distances between data points in feature spaces; but, using Euclidean distances to cluster data usually does not work effectively. Inspired by Xie [9], this paper proposes a method for simultaneously representing and clustering texts. This method uses the CNN model to express the text into a new feature space for optimizing the clustering target, and uses the gradient descent method of back propagation to learn CNN parameters based on the clustering target.

2 Model Introduction

Consider the problem of clustering a set of points $\{x_i \in X\}_{i=1}^{n}$ into k clusters (each cluster represented by its center $\mu_j, j=1,...,k$). Instead of clustering directly in data space X, this paper uses a nonlinear mapping $f_\theta : X \to Z$ to transform data. θ is the learnable parameter, Z is the latent feature space, and the dimension of Z is usually much smaller than X. In order to parametrize f_θ, deep neural networks become natural choices due to their nonlinear and characteristic learning ability.

This paper proposes an algorithm that simultaneously learns k cluster centers and deep neural network parameter θ in the clustering space. The algorithm proposed in this paper is divided into two steps: (1) parameter initialization using CNN network; (2) parameter optimization. In the following, this paper describes the two steps separately.

2.1 Parameter Initialization

In this paper, the dynamic convolutional network (DCNN) with local reservation constraints proposed by Xu et al. [10] is used to perform the parameter initialization of the CNN model. The DCNN can transform the original texts into a valid representation. Let $X = \{x_i : x_i \in R^{d\times 1}\}_{i=1,2,...,n}$ represents the input n documents, where d is the dimension of the original keyword feature. Each original text vector x_i is mapped to the matrix representation $S \in R^{d_w \times s}$ by the query word embedding E, where d_w is the dimension of the word embedding feature, s is the length of the text. Let $\tilde{W} = \{W_i\}_{i=1,2}$ and W_O represent the weight of the neural network. The network defines a transformation $f(\cdot): R^{d\times 1} \to R^{r\times 1}$ $(d \gg r)$, mapping from the original input text x to the r dimensional depth representation h. The network has three basic operations: wide one-dimensional convolution, folding, and dynamic k-max pooling. 2 convolution layers are sets up in the network by experiments; and a folding layer is added after each convolution layer. The width of the convolution filter is set to 3; the parameter k in the k-max pooling is set to 5; the word embedding dimension is set to 45; and r is set to 500.

First, train the B code based on the keyword features with the local reservation constraints. The final layer of CNN is as follows:

$$O = W_O h \tag{1}$$

Where h is the depth feature representation, $O = W_O h$ is the output vector, and $W_O \in R^{q\times r}$ is the weight matrix. To train the B code, apply the q logistic operation on the output vector O:

$$p_i = \frac{\exp(O_i)}{1+\exp(O_i)} \tag{2}$$

Given the training text set X and pre-trained B code, all parameters that need training can be defined as θ:

$$\theta = \{E, \tilde{W}, W_O\} \tag{3}$$

Use the method proposed in [11] to train the CNN network, the likelihood function of the parameter

group is:

$$J(\theta) = \sum_{i=1}^{n} \log p(b_i \mid x_i, \theta) \tag{4}$$

After the training, the local reservation constraints of the CNN network is removed, and the preserved data is taken as the initial mapping from the data space to the feature space, which is shown in Fig. 1. In order to initialize the clustering center, the data is passed to the initial CNN to get the embedding points, and the initial cluster centers $\{\mu_j\}_{j=1}^{k}$ are obtained by using the standard k-means algorithm in the feature space Z.

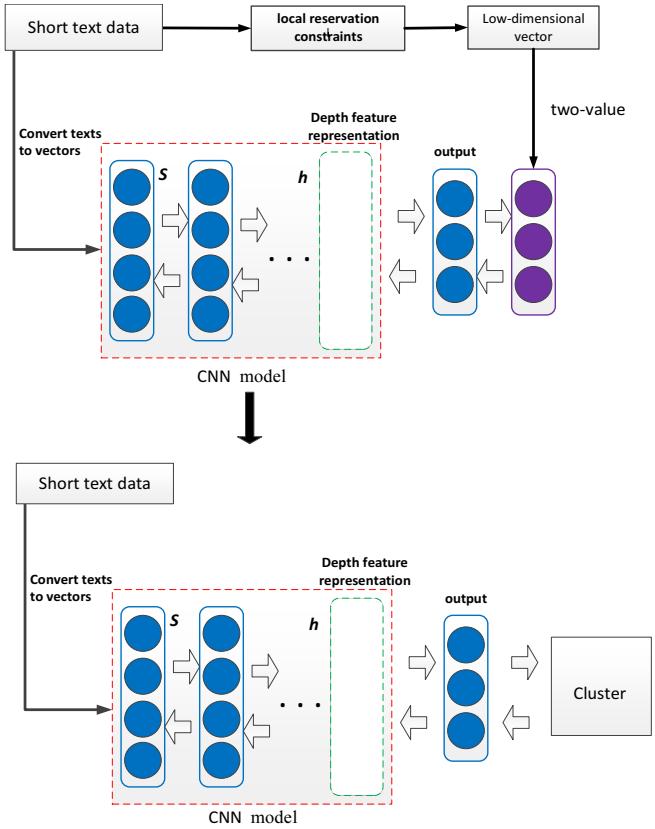

Fig. 1. CNN-based short text representation and clustering model

2.2 Using KL divergence clustering

Given the initial estimation of the nonlinear mapping f_θ and the initial cluster centers $\{\mu_j\}_{j=1}^{k}$, this paper uses an unsupervised clustering method that iterates between two steps. In the first step, we calculate the soft allocation between the embedding points and the clustering centers. In the second step, we update the depth map f_θ and learn the clustering centers from the high confidence allocation by using the auxiliary target distribution. This process continues until the convergence conditions are met.

Soft allocation. This paper uses t distribution as the kernel to embed the distance between the point z_i and the cluster center μ_j,

$$q_{i,j} = \frac{(1 + \left\| z_i - \mu_j \right\|^2 / \alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'} (1 + \left\| z_i - \mu_{j'} \right\|^2 / \alpha)^{-\frac{\alpha+1}{2}}} \tag{5}$$

Where $z_i = f_\theta(x_i)$ corresponds to $x_i \in X$ after embedding, α is the degree of freedom of t

distribution, and q_{ij} can be regarded as the probability of allocating sample i to cluster j. Refer to ref. [12], we set the parameter $\alpha = 1$ in this paper.

KL divergence minimization. This paper uses an auxiliary target distribution method to learn clustering centers from high confidence allocations. The model is trained by matching the soft allocation and the target distribution. Thus, the KL divergence between the soft allocation q_i and the auxiliary target distribution can be used as the objective function:

$$L = KL(P \| Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{6}$$

The choice of target distribution P is very important for the representations of the clustering method in this paper. A simple method is to set each distribution p_i as a delta distribution, so only those data points above the confidence threshold are retained and the rest are discarded. However, because of p_i belonging to a soft allocation, a more "softened" probability target should be used. The probability target meeting the requirements should satisfy the following characteristics: (1) enhance the prediction effect, (2) focus on high confidence data points, and (3) normalize the loss function of each centroid to prevent large clusters interference feature spaces.

This paper calculates the value of p_i by squaring q_i and using the frequency normalization of each cluster:

$$p_{i,j} = \frac{q_{ij}^2 / f_j}{\sum_{j'} q_{ij'}^2 / f_{j'}} \tag{7}$$

Where $f_j = \sum_i q_{ij}$ is the soft clustering frequency.

The training strategy in this paper can be regarded as a self-training strategy [13]. For the self-training strategy, first an initial classifier and an unlabeled data set are given, then a classifier is used to annotate the data set and train on a high confidence prediction.

Optimization. This paper uses momentum random gradient descent to optimize cluster centers $\{\mu_j\}$ and CNN parameter θ. L the gradients for each data point z_i in the feature space and the gradient of each cluster center μ_j are as follows:

$$\frac{\partial L}{\partial z_i} = \frac{\alpha+1}{\alpha} \sum_j (1 + \frac{\left\| z_i - \mu_j \right\|^2}{\alpha})^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j) \tag{8}$$

$$\frac{\partial L}{\partial \mu_j} = -\frac{\alpha+1}{\alpha} \sum_i (1 + \frac{\left\| z_i - \mu_j \right\|^2}{\alpha})^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j) \tag{9}$$

Then the gradient $\partial L / \partial z_i$ is passed to the CNN model, and the standard back propagation is used to calculate the parameter gradient $\partial L / \partial \theta$ of the CNN model. The iteration is stopped when less than $tol\%$ of data points change the clusters between two consecutive iteration steps.

3 Experiment

3.1 Data Set

This paper tests the proposed algorithm on two datasets. The two datasets are as follows:

SearchSnippets. This dataset is a result from web searches using pre-defined phrases from 8 different domains.

Reuters. This dataset contains 810,000 English news items marked with category trees. This paper uses four root types: enterprise/industry, government/society, market, and economy. And all documents that have been tagged by multiple root nodes are removed.

3.2 Evaluation Indicators

This paper evaluates the clustering performance by comparing the text clustering results with the tags provided by the text corpora. This paper uses two indicators, Accuracy (ACC) and Normalized Mutual Information (NMI) to evaluate the clustering performance. Given text x_i, let c_i and y_i represent the obtained class label and the label provided by the corpora library, respectively. The definition of accuracy is as follows:

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i, map(c_i))}{n} \tag{10}$$

Where n is the total number of texts, $\delta(x, y)$ is an indicator function equaling to 1 If $x = y$ and equaling to 0 in other cases, $map(c_i)$ is an one-to-one mapping between clusters and tags.

3.3 Algorithm Comparison

The following algorithms are selected to compare with the algorithm proposed in this paper:
K-means. This method is a traditional unsupervised clustering method. In this paper, the original keyword features are first weighted by word frequency-inverse document frequency (TF-IDF) and then clustered by the K-means algorithm.
Spectral clustering. The method uses the Laplasse characteristic graph (LE) first and then the K-means algorithm.
Average embedding. This method weight word embedding with TF-IDF first and then uses K-means algorithm for clustering.
DCNN using K-means. This method uses the K-means algorithm to cluster the proposed CNN network training results.

3.4 Performance Comparison

Table 1 below shows the performance of the indicator ACC on two datasets for various clustering algorithms.

Table 1. Algorithm performance comparison

Corpus	SearchSnippets	Reuters
K-means	27.34	25.67
Spectral clustering	65.23	59.71
Average embedding	57.83	48.70
DCNN (K-means)	72.69	68.13
Proposed algorithm	74.93	72.22

Experiments show that the results of spectral clustering and average embedding are better than K-means. This is mainly because K-means constructs similarity features directly from original keyword features and spectral clustering and average embedding use shallow feature models to obtain semantic feature models. DCNN (K-means) model transforming the original texts into a deep semantic feature space achieves better results than K-means, spectral clustering and average embedding algorithms. The algorithm presented in this paper optimizes the clustering targets while learning feature representations and clustering. It achieves better results than the benchmark algorithm on the used dataset.

4 Conclusion

This paper proposes a method to process text representations and clustering simultaneously. In the first stage, the method uses the dynamic convolutional network (DCNN) with local reservation constraints to

initialize the parameters of the CNN model; in the second stage, the method uses an unsupervised clustering method that iterates between two steps to learn the text representations and clustering targets by deep neural network. The results on the test dataset show that the proposed method has a certain degree of improvement compared with the benchmark algorithm.

Acknowledgements

This work was partially supported by 2018 Shenzhen Discipline Layout Project (No. JCYJ20170815 145900474), and Shenzhen Basic Research Project (No. JCYJ20170818115704188).

References

[1] P. Ferragina, U. Scaiella, Fast and accurate annotation of short texts with Wikipedia pages, Software IEEE 29(1)(2010) 70-75.

[2] V. Pandy, Integrating ontology to enhance HCL-based text document clustering, Research Journal of Applied Sciences 8(7)(2013) 358-368.

[3] J. Yin, J. Wang, A dirichlet multinomial mixture model-based approach for short text clustering, in: Proc. the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014.

[4] D. Cai, X. He, J. Han, Document clustering using locality preserving indexing, IEEE Transactions on Knowledge & Data Engineering 17(12)(2005) 1624-1637.

[5] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313(5786)(2006) 504-507.

[6] Y. Goldberg, O. Levy, word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. <http://arxiv.org/abs/1402.3722>, 2014.

[7] J. Cheng, D. Kartsaklis, E. Grefenstette, Investigating the role of prior disambiguation in deep-learning compositional models of meaning. <http://arxiv.org/abs/1411.4116>, 2014.

[8] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougaras, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation. <http://arxiv.org/abs/1406.1078>, 2014.

[9] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: Proc. International Conference on Machine Learning (ICML), 2016.

[10] J. Xu, P. Wang, G. Tian, B. Xu, J. Zhao, F. Wang, H. Hao, Short text clustering via convolutional neural networks, in: Proc. NAACL-HLT, 2015.

[11] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences. <http://arxiv.org/abs/1404.2188>, 2014.

[12] L.V.D. Maaten, Learning a parametric embedding by preserving local Structure, Journal of Machine Learning Research 5(2009) 384-391.

[13] K. Nigam, R. Ghani, Analyzing the effectiveness and applicability of co-training, in: Proc. the Ninth International Conference on Information and Knowledge Management, 2002.